



biblio.ugent.be

The UGent Institutional Repository is the electronic archiving and dissemination platform for all UGent research publications. Ghent University has implemented a mandate stipulating that all academic publications of UGent researchers should be deposited and archived in this repository. Except for items where current copyright restrictions apply, these papers are available in Open Access.

This item is the archived peer-reviewed author-version of:

Distributed Video Coding with Feedback Channel Constraints

Jürgen Slowack, Jozef Škorupa, Nikos Deligiannis, Peter Lambert, Adrian Munteanu, and Rik Van de Walle

In: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22 (7), pp.1014-1026, 2012.

DOI: [10.1109/TCSVT.2012.2189669](https://doi.org/10.1109/TCSVT.2012.2189669)

To refer to or to cite this work, please use the citation to the published version:

Jürgen Slowack, Jozef Škorupa, Nikos Deligiannis, Peter Lambert, Adrian Munteanu, and Rik Van de Walle (2012). Distributed Video Coding with Feedback Channel Constraints. *IEEE Transactions on Circuits and Systems for Video Technology*. Vol.22 (7), pp. 1014-1026. doi: [10.1109/TCSVT.2012.2189669](https://doi.org/10.1109/TCSVT.2012.2189669)

Distributed Video Coding with Feedback Channel Constraints

Jürgen Slowack, Jozef Škorupa, Nikos Deligiannis, Peter Lambert, Adrian Munteanu, and Rik Van de Walle

Abstract—Many of the Distributed Video Coding (DVC) systems described in the literature make use of a feedback channel from the decoder to the encoder to determine the rate. However, the number of requests through the feedback channel is often high, and as a result the overall delay of the system could be unacceptable in practical applications. As a solution, feedback-free DVC systems have been proposed, but the problem with these solutions is that they incorporate a difficult trade-off between encoder complexity and compression performance.

Recognizing that a limited form of feedback may be supported in many video streaming scenarios, in this paper we propose a method for constraining the number of feedback requests to a fixed maximum number of N requests for an entire Wyner-Ziv (WZ) frame. The proposed technique estimates the WZ rate at the decoder using information obtained from previously decoded WZ frames, and defines the N requests by minimizing the expected rate overhead. Tests on eight sequences show that the rate penalty is less than 5% when only 5 requests are allowed per WZ frame (for a GOP of size four). Furthermore, due to improvements from previous work, the system is able to perform better than or similar to DISCOVER even when up to 2 requests per WZ frame are allowed.

The practical usefulness of the proposed approach is studied by estimating end-to-end delay and encoder buffer requirements, indicating that DVC with constrained feedback can be an important solution in the context of video streaming scenarios.

Index Terms—distributed video coding, feedback channel, rate estimation

I. INTRODUCTION

DUE to practical limits on storage capacity and transmission bandwidth, video compression has always been an important field of research. Traditional video coding solutions (such as H.264/AVC) realize compression by performing a high number of computations at the encoder's side, leaving the decoder fairly simple. In contrast to these conventional solutions, distributed video coding (DVC) has emerged during the past decade as a new video coding paradigm, shifting the complexity from the encoder to the decoder's side.

One of the most popular DVC architectures is the architecture developed at Stanford by Aaron et al. [1], [2]. In this system, first, frames are partitioned into key frames

and Wyner-Ziv (WZ) frames. The key frames are coded and decoded without using other frames as references (e.g., using H.264/AVC intra coding). For each WZ frame, the decoder generates a prediction which is called the *side information*. To correct errors in the side information, error correcting information (such as turbo or LDPC codes) is sent by the encoder to the decoder. As such, one very important question to address is: how does the encoder know the amount of error correcting information needed by the decoder for successful decoding? After all, sending not enough information results in unsuccessful decoding, while sending too much information results in bit rate overhead.

The strategy applied in the Stanford codec and most of its extensions (such as [3]–[8] and DISCOVER [9]) is to use a feedback channel from the decoder to the encoder. Basically, it is up to the decoder to determine the rate by requesting chunks of error correcting information until the decoding process is considered reliable. As such, the decoder can avoid bit rate overhead by requesting bit chunks one by one.

Although the use of such a feedback channel leads to the highest performance reported in the literature, there are some important implications. First of all, storage applications are difficult to support since the rate of the coded stream can only be determined in a scenario involving both the encoder and the decoder. Secondly, in a practical setup there is usually a non-negligible delay associated with the forward and backward communication channel. As a result, frequent use of the feedback channel might result in end-to-end delays that are too high for practical usage scenarios.

To overcome these problems, a number of systems have been proposed in which a feedback channel is no longer present, e.g., [10]–[12] as well as the pioneering PRISM architecture [13]. Here it is the encoder that determines the rate, denying the decoder the right to issue any requests for bits. Although such systems are surely more practical, the main problem at hand is that the encoder is not allowed to perform a lot of computations, as this would interfere with the DVC paradigm of simple encoders and complex decoders. So, if the encoder should remain simple, then how should it accurately estimate the minimal amount of bits needed by the decoder?

Typically, in the literature, the problem of encoder-side rate estimation is solved by generating an estimation of the side information using very simple techniques. This estimate is then compared to the frame to be coded in order to determine the rate. For example, in PRISM [13], the rate is determined by comparing each block of pixels with the co-located block in the previous frame. More recent approaches use the average of two adjacent key frames [10], or fast motion estimation [11],

J. Slowack, J. Škorupa, P. Lambert, and R. Van de Walle are with the Multimedia Lab, Department of Electronics and Information Systems, Ghent University, Gaston Crommenlaan 8 (box 201), B-9050 Ledeberg-Ghent, Belgium.

N. Deligiannis and A. Munteanu are with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium.

J. Slowack, J. Škorupa, N. Deligiannis, P. Lambert, A. Munteanu, and R. Van de Walle are also with the Interdisciplinary Institute for Broadband Technology (IBBT), Gaston Crommenlaan 8 (box 102), B-9050 Ledeberg-Ghent, Belgium.

[12] to obtain an estimation of the side information.

One of the disadvantages of the feedback-free DVC systems is that there is a trade-off between the complexity of the encoder and the compression performance, since more advanced encoder-side rate estimation algorithms typically lead to better compression. It should also be remarked that the feedback-free systems force the encoder and decoder to be logically connected, in the sense that the techniques used at the encoder for estimating the bit rate should match the techniques applied at the decoder. As a consequence, research and improvements on the decoder's side imply modifying the encoder as well. Otherwise, the encoder would simply send the same amount of bits, unaware that the decoder is able to successfully decode using less bits than in the initial design. Another issue is that – because of the fact that the techniques for generating the side information at the decoder are expected to become more and more complex – we expect that it will become increasingly difficult to accurately estimate the bit rate using low-complexity techniques at the encoder.

A limited number of contributions have been proposed in the literature that attempt to reduce the number of requests through the feedback channel without eliminating it completely. In these systems, instead of sending only one chunk of bits at a time, the encoder first estimates an initial number of chunks to send. For example, Kubasov et al. [14], propose to estimate the initial number of chunks by using knowledge about the correlation between the side information and the original frame. Instead, Areia et al. [15] propose to exploit knowledge about the bit rates spent in previously coded frames. After sending the initial number of chunks, both approaches revert to the conventional strategy in which the decoder requests additional information until decoding is successful. While these techniques indeed decrease the number of requests through the feedback channel with a limited impact on performance, still, the feedback channel is left unconstrained. Therefore, the delay of the system may still be too high or too much varying for practical scenarios.

The solution we propose in this paper is to use *constrained* feedback. We believe that – apart from storage applications – it is realistic to assume that a feedback channel is present in practice. However, the system should be able to cope with its limitations in terms of delay. In our setup, the decoder is only allowed to issue N requests for bits to the encoder, for the whole WZ frame. One of the advantages of this strategy is that the decoder can be responsible for determining the bit rate, hereby avoiding the trade-off between encoder-side complexity and compression performance as in the case of feedback-free DVC architectures. Instead, compression is influenced by N in the sense that higher values lead to better performance. This tradeoff is most likely easier to address since network latency (and hence N) is expected to improve (or at least remain the same) in the future. As a second important advantage, there is no logical connection between the encoder and the decoder, meaning that gain can be achieved by modifying only the decoder. This could be important in the context of deployment and/or standardization.

The structure of this paper is as follows. In Section II we discuss the architecture used as a starting point in this paper.

In Section III we comment on the problems associated with current sequential decoding approaches. Next, techniques are developed that enable adapting to a fixed constraint on the number of feedback requests, as described in Section IV. The efficiency of these techniques is evaluated in Section V. In Section VI we describe how such systems may operate in practice, and we analyze important properties such as end-to-end delay and encoder storage requirements. Conclusions and future work end the paper in Section VII.

II. GENERAL CODEC OPERATION

The starting point of this paper is the codec described in our previous work [16]. We will introduce this system from a high-level point of view, providing all the details necessary to understand the contributions in this paper. For the remaining details, the reader is referred to the literature [16].

Fig. 1 depicts the architecture of the codec, which is based on the system initially proposed by Aaron et al. [2] with some important extensions in the context of side information generation, correlation noise estimation, and mode decision. We will first describe the operation of the encoder in Section II-A. This is then followed by a description of the decoder in Section II-B.

A. Encoder operation

At the encoder, the frame sequence is partitioned into key frames I and WZ frames W . A (fixed) hierarchical GOP structure is used, meaning for example that the sequence $I_1 - W_2 - W_3 - W_4 - I_5$ is coded and decoded in the following order: $I_1 - I_5 - W_3 - W_2 - W_4$. First, the key frames (I_1 and I_5) are coded using H.264/AVC intra coding techniques. Next, the middle WZ frame W_3 is partitioned into non-overlapping blocks of size 4×4 , and each of these blocks is transformed using the discrete cosine transform (DCT). Transform coefficients X_k at the same index k ($0 \leq k \leq 15$) are grouped into so-called coefficient bands. For example, all DC coefficients in W_3 form coefficient band zero. Each coefficient band is quantized using a quantizer having 2^{M_k} or $2^{M_k} - 1$ levels, i.e., for the DC coefficient band or AC coefficient bands, respectively. For the AC coefficients, deadzone quantization is performed, meaning that the center quantization bin (containing zero) is 1.5 the size of the other bins. After quantization, for each coefficient band the bits at corresponding positions are grouped into bitplanes. For example, all most significant bits from the first coefficient band are grouped into one bitplane.

Each bitplane is either (1) skipped, (2) intra coded using binary arithmetic coding, or (3) WZ coded using a turbo coding strategy (which is the conventional DVC approach). Mode decision is performed at the decoder side, and the encoder is notified of the mode through the feedback channel, hereby operating the switch denoted S .

B. Decoder operation

After decoding the key frames I_1 and I_5 , the decoder generates side information for W_3 through motion compensated

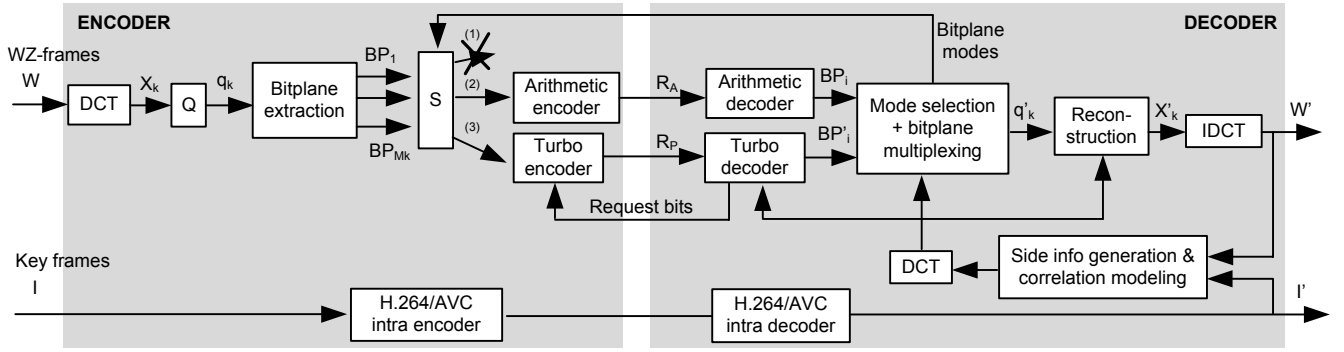


Fig. 1. The DVC codec presented in our previous work [16], featuring several modes for coding bitplanes, i.e., (1) bitplane skip, (2) bitplane intra coding, and (3) bitplane WZ coding. This codec is used as a starting point in this paper.

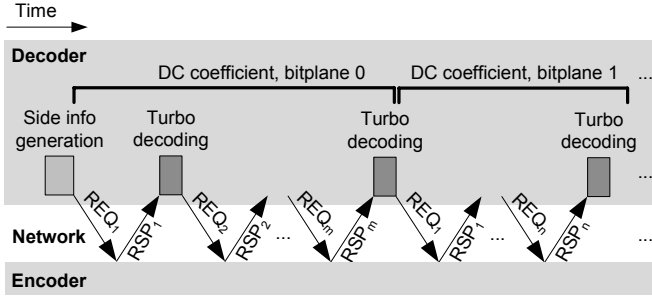


Fig. 2. In a conventional DVC feedback scenario, bitplanes in the same coefficient band are decoded sequentially, and for each bitplane the number of requests is unconstrained.

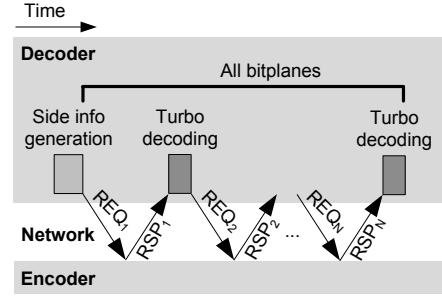


Fig. 3. In the proposed scheme, the decoder is only allowed to communicate information to the encoder N times per WZ frame. Each request and response can contain information about multiple bitplanes.

interpolation. In our system, we have adopted the techniques for side information generation as used in DISCOVER [9]. The decoder also estimates the correlation between Y and W_3 , using techniques from previous work [17].

Using the correlation model and information from previously decoded frames, the decoder determines on a rate-distortion basis which mode should be used for coding a certain bitplane. (1) If the side information bitplane is considered sufficiently reliable, the decoder notifies the encoder to skip this bitplane. In that case, the side information bitplane is considered the result. Otherwise, the decoder decides upon the best mode to use, i.e., (2) the intra mode or (3) the WZ mode, based on the results from previously decoded frames.

After all bitplanes have been decoded, the bitplanes are multiplexed and the coefficients are reconstructed through centroid reconstruction. The result is inverse transformed to obtain the decoded WZ frame W'_3 . This frame can be used for generating side information for the other WZ frames to be decoded (such as W_1 etc.).

III. PROBLEMS WITH CURRENT SEQUENTIAL DECODING METHODS

The architecture described in the previous section adopts a feedback strategy for communicating mode information and for issuing parity bit requests. For the latter, it is important to remark that feedback-based DVC systems typically use information from previously decoded bitplanes to decode the current bitplane in the same coefficient band [18]. As a result,

feedback requests are issued in a sequential way, as illustrated in Fig. 2.

Other researchers have extended this sequential way of decoding. For example, in addition to bitplane-level sequential decoding, Martins et al. [5] propose coefficient band sequential decoding to allow refining the side information for the following coefficient bands to be decoded. Similar refinement techniques have been proposed in other contributions, at coefficient band or bitplane level [19]–[21], or using spatial layers [22].

Although sequential decoding and refinement has shown to significantly improve compression, supporting these techniques through sequential feedback requests is difficult, particularly when constraining the total number of requests to a feasible value. After all, the total number of bitplanes in one WZ frame is quite large, e.g., ranging from 10 to 63 bitplanes [23]. If a feedback channel is used for each of these bitplanes in a sequential manner, the delay could be too high for use in practice.

Therefore, we develop a scheme in which requests for multiple bitplanes are concatenated into one single message sent to the encoder. This idea is illustrated in Fig. 3: each request REQ_i and response RSP_i (with $1 \leq i \leq N$) contains information about multiple bitplanes, in order to ensure successful decoding after a limited number of requests N .

IV. PROPOSED TECHNIQUE

The scheme depicted in Fig. 3 is realized by modifying the DVC decoder discussed in Section II-B. The operation of the encoder remains the same. The different steps performed at the decoder for decoding a WZ frame are summarized in Fig. 4, and discussed next.

Based on the characteristics of the network, the decoder first determines a feasible value for N . This is discussed in Section VI, after we have described the other steps in detail. After generating the side information and modeling the correlation, the decoder estimates the rate required for intra coding and WZ coding each bitplane, as described in Section IV-A1 and Section IV-A2, respectively. The accuracy of the WZ rate estimation process is modeled (Section IV-B), to allow defining each of the N requests through minimization of the expected rate overestimation (Section IV-C). Next, based on the results of the rate estimation process, the decoder decides upon the mode to use for each of the bitplanes, as explained in Section IV-D.

Once all bitplane modes and WZ requests are defined, the decoder proceeds as depicted in Fig. 3 by sending a first message to the encoder. This request REQ_1 contains the modes for all bitplanes, as well as the desired first amount of parity bits for the WZ coded bitplanes. Unless all bitplanes are skipped, the encoder responds by sending the intra coded bitplanes (if any) and the requested number of parity bits for the WZ bitplanes (if any). This response is denoted RSP_1 . The decoder interprets this response, decodes the intra coded bitplanes, and runs the turbo decoder for the WZ bitplanes. If WZ decoding is unsuccessful for one or more bitplanes, additional bits are requested in a second request REQ_2 , and turbo decoding is restarted upon receiving the response. This step is repeated until all bitplanes are reliably decoded or until N requests have been issued.

After decoding, the decoder recalculates the minimal number of WZ bits that would have been required for the WZ bitplanes, as described in Section IV-F. This information is used in the context of rate estimation for subsequent WZ frames to be decoded. Finally, the bitplanes are combined and the transformation coefficients are reconstructed (as shown in Section IV-G), in order to obtain the decoded WZ frame after applying the inverse DCT.

We will now describe each step in detail. Fig. 4 could be used as a guide for the reader to maintain a good view on the overall workflow.

A. Bitplane rate estimation

After generating the side information for a particular WZ frame and estimating the correlation, the following step at the decoder is estimating the rate required to decode each bitplane, both in the case of intra coding (Section IV-A1) and in the case of WZ coding (Section IV-A2).

1) *Intra mode*: A straightforward approach is used to estimate the number of intra bits required for coding a certain bitplane: after transforming the side information using the DCT, it is quantized and the resulting bitplanes are extracted. Next, each of the bitplanes is intra coded, using the same

DECODER OPERATIONS FOR EACH WZ FRAME

- Determine N , based on network characteristics and encoder constraints (Section VI).
- Generate side information through motion-compensated interpolation [9].
- Estimate correlation between side information at the decoder and original at the encoder [17].
- For each bitplane,
 - estimate intra rate (Section IV-A1),
 - estimate WZ rate (Section IV-A2),
 - model accuracy of WZ rate estimation (Section IV-B),
 - define the N WZ feedback requests allowed (Section IV-C),
 - perform mode decision, i.e., decide between skip, intra, and WZ coding (Section IV-D),
- Execute requests (as described in Fig. 3), i.e.,
 - 1) send REQ_1 to the encoder, containing bitplane modes and WZ rate requests,
 - 2) upon receiving RSP_1 , decode intra bitplanes,
 - 3) run turbo decoder for WZ bitplanes (Section IV-E),
 - 4) if WZ decoding is not successful, issue a new request for bits,
 - 5) upon receiving the encoder's response, run turbo decoder again for WZ bitplanes,
 - 6) if less than N requests have been issued, go to 4.
- After bitplane decoding:
 - calculate minimal required WZ rate for each bitplane (Section IV-F),
 - group bitplanes and perform coefficient reconstruction (Section IV-G).

Fig. 4. Overview of decoder-side computations required to support WZ decoding with constrained feedback. The details of the different steps are described in the literature, or in this paper, as indicated.

binary arithmetic coder as the one that will be used at the encoder. The resulting amount of intra coded bits serves as an estimation of the intra rate that would be spent on intra coding the original bitplane (available at the encoder).

The accuracy of this technique is acceptable, as illustrated by Fig. 5 for two examples.

2) *Wyner-Ziv mode*: To estimate the WZ rate we exploit knowledge about the rates required for previously decoded WZ frames. To account for variations in rate due to differing distances between the reference frames used for side information generation, only previously decoded frames in the same hierarchical layer are considered. For example, using a GOP of size four labeled $I_1 - W_2 - W_3 - W_4 - I_5 - W_6 - W_7 - W_8 - I_9 - \dots$, one hierarchical layer will contain WZ frames W_3, W_7, \dots (for which reference frames are four frames apart), while another layer will contain the remaining frames W_2, W_4, \dots (for which reference frames are two frames apart).

Using the concept of hierarchical layers, the WZ rate for a particular bitplane BP is determined based on the results for the collocated bitplanes (denoted BP_{-1} , BP_{-2} , and BP_{-3}) in the three previously decoded frames¹ in the same hierarchical layer. As will be explained in Section IV-F, for each of these three bitplanes the decoder has access to a very accurate post-decoding estimation of the WZ rate. This enables estimating the WZ rate R' for the current bitplane as:

$$R' = \text{med}(\tilde{R}_{-1}, \tilde{R}_{-2}, \tilde{R}_{-3}), \quad (1)$$

where med denotes the median operator, and \tilde{R}_{-1} , \tilde{R}_{-2} , and \tilde{R}_{-3} denote the post-decoding estimation of the WZ rate for the collocated bitplanes BP_{-1} , BP_{-2} , and BP_{-3} , respectively.

¹The first three frames of a video sequence are coded using only intra and skip mode.

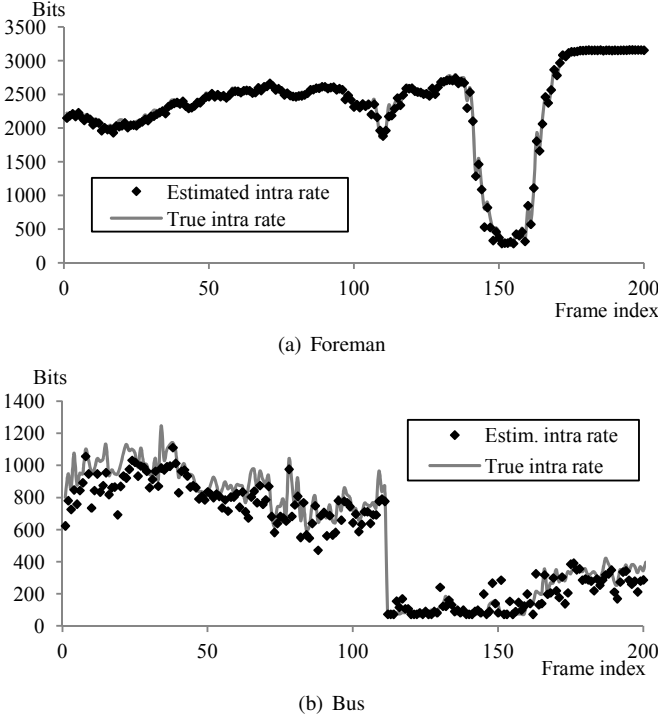


Fig. 5. Illustration of the accuracy of the intra rate estimation for the most significant bitplane of the luma DC of Foreman (CIF, 30 Hz, GOP 4, second RD point), and the fourth bitplane of the second coefficient of Bus (CIF, 30 Hz, GOP 4, third RD point).

The accuracy of this predictor is illustrated by Fig. 6.

B. Modeling WZ rate estimator accuracy

It is important to model the accuracy of the WZ rate estimation process as defined in the previous subsection. If R' is considered less accurate, it would make more sense to spread the N available requests, i.e., apply larger rate increments between requests. On the other hand, if R' is considered accurate, one could choose finer increments to minimize the probability of requesting more bits than strictly needed for decoding.

As typical in DVC, parity bits are sent in chunks to the decoder, where the number of bits in one chunk is determined by the puncturing period used. With the estimated minimal number of WZ bits R' expressed in chunks, denote R the correct minimal number of WZ chunks. Through experiments we have found that the error $R - R'$ can be accurately modeled by a zero-mean Laplace distribution (as illustrated in Fig. 7), i.e.:

$$f_{R-R'}(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \quad (2)$$

where α is the distribution scale parameter.

The α parameter is typically sequence-dependent and temporally varying. Therefore, α is estimated during decoding by interpreting the results for the M previously decoded WZ frames in the same layer. For these frames, denote the estimated WZ rate *prior* to decoding as R'_{-m} (with $1 \leq m \leq M$). These values have been calculated as described in Section IV-A2. We use the same notation as before for the

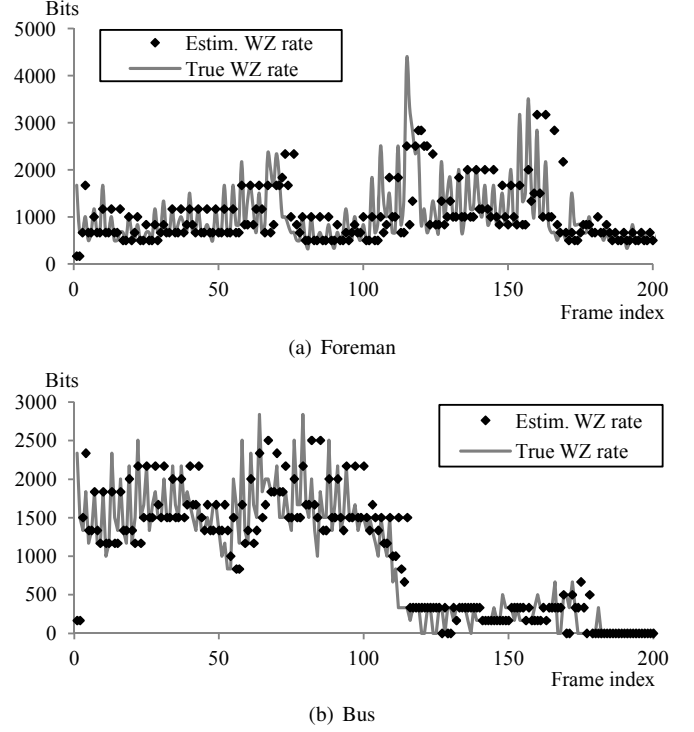


Fig. 6. Illustration of the accuracy of the WZ rate estimation process for the most significant bitplane of the luma DC of Foreman (CIF, 30 Hz, GOP 4, second RD point), and the fourth bitplane of the second coefficient of Bus (CIF, 30 Hz, GOP 4, third RD point).

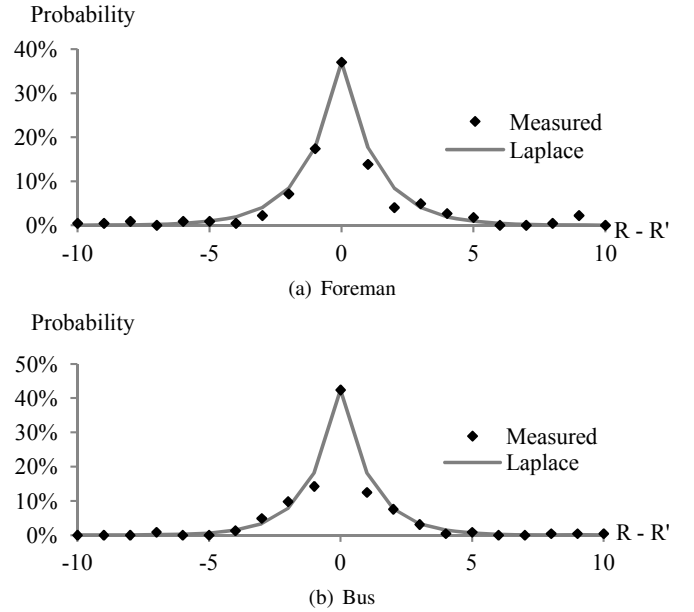


Fig. 7. Measured distribution of $R - R'$ fitted by a Laplacian, for the most significant bitplane of the luma DC of Foreman (CIF, 30 Hz, GOP 4, second RD point), and the fourth bitplane of the second coefficient of Bus (CIF, 30 Hz, GOP 4, third RD point).

updated values (according to Section IV-F) *after* decoding, i.e., \tilde{R}_{-m} .

Using these notations, α is estimated by fitting a Laplace distribution to the error samples $\tilde{R}_{-m} - R'_{-m}$. As a fitting method, we adopt maximum likelihood fitting, delivering the following expression for α :

$$\alpha = \frac{1}{\frac{1}{M} \sum_{m=1}^M |\tilde{R}_{-m} - R'_{-m}|}. \quad (3)$$

The number of error samples M relates to the sensitivity of the decoder to temporal inaccuracies in the rate estimation process. If M is chosen too small, the fitted distribution will not be statistically relevant. On the other hand, if M is chosen too large, the system will not be sensitive enough to adapt to short temporal changes in the accuracy of the rate estimation process. Based on experiments, $M = 10$ showed to provide a good balance between adaptiveness and relevance, and so this value will be used in this paper.

C. Defining the rate requests

Given R' and its estimated error distribution $f_{R-R'}$, the main problem left at this moment is how to define the number of bit chunks to request for each of the maximum N requests.

In case a particular WZ bitplane cannot be decoded after N requests, the bits requested so far for this bitplane are essentially overhead since they did not contribute positively to the turbo decoding process applied at the decoder. This uncorrected bitplane increases distortion for the current frame, as well as for other WZ frames using this frame as a reference frame for side information generation. On the other hand, in the case of rate overestimation the penalty remains limited to bit rate overhead only. In addition, if decoding succeeds, the minimal number of bit chunks can be retrieved easily, as will be described further on.

Due to these reasons, we prefer to use a rate request strategy that avoids underestimating the number of bit chunks at all times. Denote, upon receiving RSP_i , the total number of WZ bit chunks received so far for a particular bitplane as R_i^* . To avoid unsuccessful decoding in the end, the number of bits received at the final response should be sufficient. Therefore, R_N^* is defined so that there is only a marginal probability ϵ for underestimating the rate. This leads to the following condition:

$$F_{R-R'}(R_N^* - R') = 1 - \epsilon, \quad (4)$$

where $F_{R-R'}$ denotes the cumulative distribution function of $f_{R-R'}$. Using the expression for the cumulative distribution function of a Laplace distribution:

$$\begin{aligned} F_{R-R'}(R_N^* - R') \\ = 0.5 \left[1 + \text{sign}(R_N^* - R') (1 - e^{-\alpha |R_N^* - R'|}) \right] \end{aligned} \quad (5)$$

in combination with Equation 4 results in the following expression for R_N^* (assuming $\epsilon < 0.5$ so $R_N^* > R'$):

$$R_N^* = R' - \frac{\ln 2\epsilon}{\alpha}. \quad (6)$$

TABLE I
FOR EACH BITPLANE, THE DECODER USES THE FOLLOWING EXPRESSIONS
TO DETERMINE THE NUMBER OF WZ BITS FOR EACH OF THE N
REQUESTS.

N = 2	
$R_1^* =$	$0.0012\sigma_{R-R'}^6 - 0.0296\sigma_{R-R'}^4 + 0.3847\sigma_{R-R'}^2 + 0.4412$
$R_2^* =$	$R' - \frac{\ln 2\epsilon}{\alpha}$
N = 3	
$R_1^* =$	$0.0003\sigma_{R-R'}^6 - 0.0076\sigma_{R-R'}^4 + 0.0986\sigma_{R-R'}^2 + 0.1131$
$R_2^* =$	$0.0020\sigma_{R-R'}^6 - 0.0517\sigma_{R-R'}^4 + 0.6719\sigma_{R-R'}^2 + 0.7704$
$R_3^* =$	$R' - \frac{\ln 2\epsilon}{\alpha}$
N = 4	
$R_1^* =$	$-0.0003\sigma_{R-R'}^6 + 0.0067\sigma_{R-R'}^4 - 0.0864\sigma_{R-R'}^2 - 0.0989$
$R_2^* =$	$0.0008\sigma_{R-R'}^6 - 0.0199\sigma_{R-R'}^4 + 0.2587\sigma_{R-R'}^2 + 0.2967$
$R_3^* =$	$0.0025\sigma_{R-R'}^6 - 0.0620\sigma_{R-R'}^4 + 0.8056\sigma_{R-R'}^2 + 0.9238$
$R_4^* =$	$R' - \frac{\ln 2\epsilon}{\alpha}$
N = 5	
$R_1^* =$	$-0.0008\sigma_{R-R'}^6 + 0.0207\sigma_{R-R'}^4 - 0.2686\sigma_{R-R'}^2 - 0.3082$
$R_2^* =$	$0.0002\sigma_{R-R'}^6 - 0.0059\sigma_{R-R'}^4 + 0.0764\sigma_{R-R'}^2 + 0.0875$
$R_3^* =$	$0.0012\sigma_{R-R'}^6 - 0.0304\sigma_{R-R'}^4 + 0.3948\sigma_{R-R'}^2 + 0.4526$
$R_4^* =$	$0.0028\sigma_{R-R'}^6 - 0.0707\sigma_{R-R'}^4 + 0.9179\sigma_{R-R'}^2 + 1.0524$
$R_5^* =$	$R' - \frac{\ln 2\epsilon}{\alpha}$

In our work, we take ϵ equal to 0.1%. This value is very low to ensure the bit rate is not underestimated, even in cases where $f_{R-R'}$ might be modeled less accurately.

With this constraint on the final request, the $N-1$ remaining requests for bit chunks are defined so that the expected rate overestimation is minimized:

$$\arg \min_{\{R_1^*, \dots, R_{N-1}^*\}} \sum_{i=1}^N \int_{R_{i-1}^*}^{R_i^*} f_{R-R'}(x - R') \cdot (R_i^* - x) dx, \quad (7)$$

where R_0^* is defined zero. Remark that only an integer number of chunks greater than zero can be requested, hence,

$$R_i^* \in \mathbb{N}, 1 \leq i \leq N \quad (8)$$

$$0 < R_1^* < R_2^* < \dots < R_{N-1}^* < R_N^*. \quad (9)$$

Finding an analytical solution to this constrained optimization problem is not straightforward, since it requires optimizing towards $N-1$ variables. In absence of a general solution, we have used numerical optimization techniques to obtain a solution for different values of N . Instead of performing numerical optimization during decoding, we have determined the optimum values in an offline setting for different N and different values of $\sigma_{R-R'}^2$. The results were fitted with third order polynomials (on $\sigma_{R-R'}^2$), as illustrated in Fig. 8. Consequently, the decoder uses only these polynomials (listed in Table I) to define each of the N requests for WZ bits.

For example, rounded to integer solutions the result for $N = 5$ and $\sigma_{R-R'}^2 = 2$ is given by: $R_1^* = R' - 1$, $R_2^* = R'$, $R_3^* = R' + 1$, and $R_4^* = R' + 3$, and $R_5^* = R' + 6$.

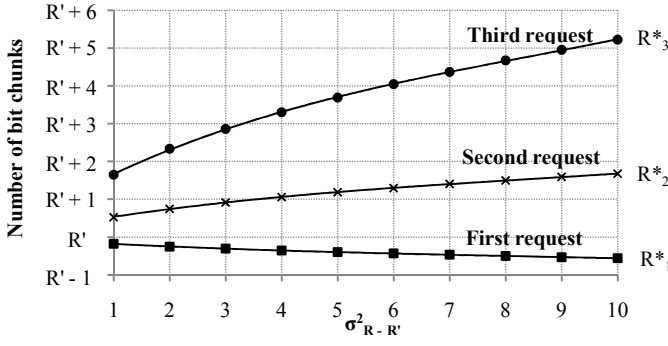


Fig. 8. Offline numerical optimization results (markers) fitted using third order polynomials on $\sigma^2_{R-R'}$ (lines), in the case $N = 4$. The final request is given by Eq. 6. Results for $N = 2$ up to $N = 5$ are listed in Table I.

D. Performing mode decision

As described in Section II, three bitplane coding modes are supported by the system, namely: skip, intra, and WZ mode. Mode decision is performed at the decoder in two stages: at the coefficient level (Section IV-D1) and at the bitplane level (Section IV-D2).

1) *Coefficient-level mode decision:* At the coefficient level, the decoder decides whether or not to skip an entire coefficient band from decoding. This step is identical to the techniques described in our previous work [16]. In essence, for a coefficient at index i in the band, two (Lagrangian) costs are calculated: one for the case of skipping the coefficient (denoted C_{skip}^i) and one for WZ coding the coefficient (denoted C_{WZ}^i). An entire coefficient band is then skipped in case $C_{skip}^i \leq C_{WZ}^i, \forall i$.

If a coefficient band is skipped, the decoder proceeds with the following band. If it is not skipped, bitplane-level mode decision is performed as described in the following section.

2) *Bitplane-level mode decision:* If a coefficient band is not skipped, the decoder tries to reconstruct all bitplanes with high reliability. Each coefficient is marked as relevant or non-relevant, depending on whether the condition $C_{skip}^i \leq C_{WZ}^i$ is true or not for this coefficient. If all relevant bits are at least 90% likely, the bitplane is skipped. Otherwise, the decoder makes a decision between intra and WZ coding using the results from the rate estimation techniques described in this paper.

For example, the decoder will spend a total of R_2^* bit chunks for a particular bitplane only in case decoding using R_1^* bit chunks failed, while decoding using R_2^* chunks succeeded. This corresponds to a probability of $F_{R-R'}(R_2^* - R') - F_{R-R'}(R_1^* - R')$. Using similar reasoning for the other requests, the expected number of bit chunks E_{WZ} used for decoding becomes:

$$\sum_{i=1}^N R_i^* \cdot (F_{R-R'}(R_i^* - R') - F_{R-R'}(R_{i-1}^* - R')) + R_N^* \cdot (1 - \epsilon), \quad (10)$$

with $F_{R_0-R'}$ defined equal to zero.

There is no rate request scheme for intra coded bitplanes, and therefore the expected number of intra bits E_{intra} is given by the techniques proposed in Sect. IV-A1.

Due to the fact that bitplane intra and bitplane WZ coding are highly likely to produce the same decoded result (and hence the same distortion), the decoder is able to decide upon the best bitplane coding mode based on the minimum between E_{WZ} and E_{intra} .

E. Feedback-based WZ decoding

Given the modes for each bitplane and the number of bits for each of the N requests, the decoder sends a first request REQ_1 to the encoder. This request contains information for all bitplanes. Upon receipt of the encoder's response, the decoder decodes the intra coded bitplanes first, and determines the bit reliabilities for the WZ bitplanes using the correlation model and the already decoded bitplanes. If WZ decoding is unsuccessful for one or more WZ bitplanes, a second request is issued, and so on.

F. Re-estimating the WZ rate after decoding

Once bitplane decoding terminates, the minimal number of WZ bit chunks can be determined². Since the complexity of the decoder is typically considered less of an issue in DVC, each decoded bitplane can be coded and decoded iteratively until the minimal number of WZ chunks has been determined. Obviously, this is only possible in case bitplane decoding was successful, but this is very likely as guaranteed by our rate request scheme.

G. Coefficient reconstruction

After bitplane decoding for a particular coefficient band has terminated, the transformation coefficients within the band are reconstructed. In contrast to the conventional case where the decoder is assumed to perfectly decode all bitplanes [24], in our case perfect decoding is not guaranteed due to the implementation of a bitplane skip mode as well as (rare) decoding failures due to feedback constraints.

Therefore we interpret bitplane decoding as a process that reduces the set of possible quantization bins S containing the original x . Prior to decoding, S contains all quantization bins (e.g., a total of 8 bins when quantizing to three bits), and each time a bitplane is successfully decoded S is updated by taking out the quantization bins that are not possible anymore. For example, when the most significant bit has been successfully decoded as being one, all bins having a most significant bit of zero will be taken out of the set S .

After decoding has terminated, S will contain one or more bins, and a particular value x' is chosen as the decoded value through centroid reconstruction over S , i.e.:

$$x' = \frac{\sum_{q \in S} \int_{q^L}^{q^H} x \cdot f'_{X|Y}(x|y) dx}{\sum_{q \in S} \int_{q^L}^{q^H} f'_{X|Y}(x|y) dx}, \quad (11)$$

where the conditional distribution $f'_{X|Y}(x|y)$ is obtained by estimating the correlation between the original and the side

²In case the bitplane was skipped, the estimated number of WZ bit chunks is set to zero. For all other cases, the decoder proceeds as described in this section.

TABLE II
AVERAGE BJØNTEGAARD DELTA [25] RATE INCREASE COMPARED TO
UNCONSTRAINED FEEDBACK, FOR A GOP OF SIZE FOUR.

	$N = 5$	$N = 4$	$N = 3$	$N = 2$	$N = 1$
Foreman	3.6 %	4.7 %	5.8 %	9.1 %	17.4 %
Table Tennis	4.1 %	6.4 %	7.4 %	16.1 %	34.1 %
Mother Daughter	2.8 %	4.2 %	5.4 %	12.1 %	28.2 %
Bus	0.6 %	1.1 %	1.6 %	2.7 %	5.6 %
Coastguard	1.7 %	2.6 %	3.2 %	6.5 %	14.3 %
Silent	2.2 %	4.1 %	5.9 %	11.4 %	28.1 %
Stefan	3.4 %	3.9 %	4.1 %	5.5 %	9.0 %
Mobile Calendar	0.5 %	1.6 %	2.3 %	5.8 %	15.6 %

information [17], and q^L and q^H denote the low and high border of the quantization bin q , respectively.

After all coefficients are reconstructed, the inverse DCT is applied to obtain the pixel values for the decoded frame.

V. RESULTS

Tests have been conducted on eight different sequences: Foreman, Table Tennis, Mother and Daughter, Bus, Coastguard, Silent, Stefan, and Mobile Calendar. All sequences are in CIF resolution, 30 Hz, coded with a GOP of length four. Only the luma component is coded to allow comparing with DISCOVER.

The discussion of the results is split into three parts. In Section V-A, we analyze compression performance as a function of N . In Section V-B, the proposed system is compared to the state-of-the-art in conventional video coding, i.e., H.264/AVC. Finally, the system is compared to other DVC systems found in the literature as described in Section V-C.

A. Compression performance as a function of N

Fig. 9 presents rate-distortion results for the Silent and Stefan sequences, comparing different values for N with the case in which the feedback channel is left unconstrained. Results for the entire test set are similar and summarized in Table II.

The results indicate that the performance of the system decreases as N decreases. Interestingly, there is only a limited impact (of up to 4.1 %) when N is set equal to five. When the number of requests is further decreased, the penalty also increases leading to often substantial penalties when only one or two requests are allowed.

The performance penalty of a particular N is sequence dependent. First of all, the accuracy of the proposed techniques for WZ rate estimation varies between the different test sequences, and so the penalty varies as well. Secondly, mode decision influences the results, in the sense that sequences with complex motion characteristics feature more intra modes which decreases the significance of the WZ stream. This explains why there are smaller losses for Bus and Stefan.

B. Comparing with H.264/AVC

As a state-of-the-art benchmark in conventional video compression with encoder-side motion estimation, two configurations of H.264/AVC are considered. Using the H.264/AVC reference software (JM 14.1, extended profile, one slice per

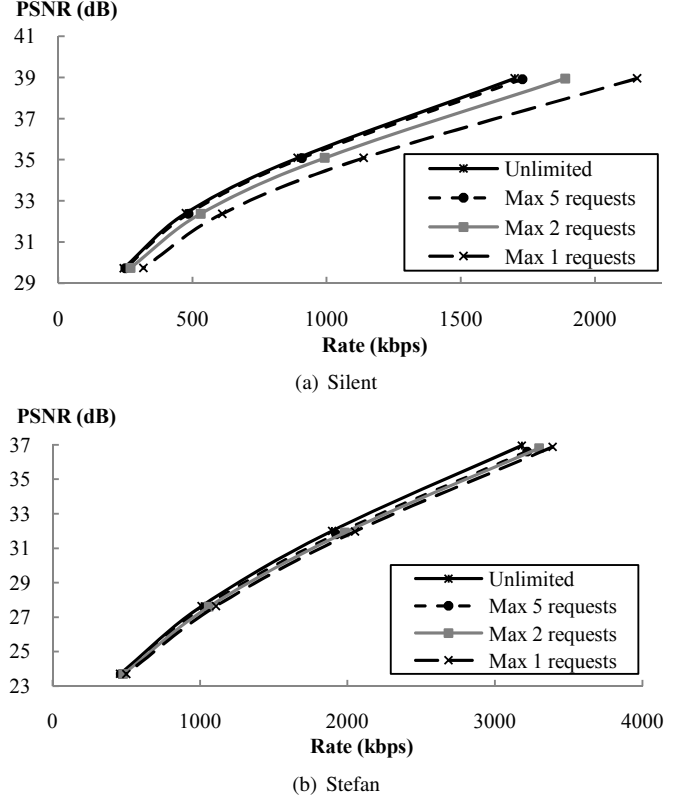


Fig. 9. RD results (CIF, 30 Hz, GOP 4) for evaluating the performance impact as a function of N . All results are summarized in Table II.

picture), two curves are generated: one for intra only and a second one applying the same hierarchical GOP coding structure as the proposed system.

Results for two sequences are provided in Fig. 10. Due to improvements from previous work (such as improved correlation noise modeling and selective bitplane intra coding [16]), the proposed system is able to outperform H.264/AVC intra coding significantly. Our experiments reveal that H.264/AVC intra is outperformed for all test sequences, even when $N = 1$. However, compared to H.264/AVC inter coding there is still quite a significant performance gap.

C. Comparing with other DVC systems

Two DVC systems are compared against the proposed solution. A first reference system is the well-known DISCOVER codec³ [9]. It is important to note that, although DISCOVER incorporates some techniques to reduce the number of feedback requests, the total number of such requests is essentially unconstrained (like in any other feedback-based DVC system described in the literature).

Table III provides full results concerning the comparison between DISCOVER and the proposed solution. Interestingly, the proposed system is able to perform on par with DISCOVER even when constrained to two requests per WZ frame. When only one request is allowed, the gap with DISCOVER is not larger than 0.8 dB.

³Executables are available online at www.discoverdvc.org. [accessed June 9, 2011]

TABLE III
AVERAGE BJØNTEGAARD DELTA [25] QUALITY IMPROVEMENT OVER DISCOVER.

	$N = 5$	$N = 4$	$N = 3$	$N = 2$	$N = 1$
Foreman	1.3 dB	1.2 dB	1.2 dB	1.1 dB	0.8 dB
Table Tennis	0.6 dB	0.5 dB	0.5 dB	0.1 dB	-0.7 dB
Mother Daughter	0.9 dB	0.8 dB	0.8 dB	0.5 dB	-0.1 dB
Bus	3.9 dB	3.8 dB	3.8 dB	3.8 dB	3.6 dB
Coastguard	0.9 dB	0.9 dB	0.8 dB	0.7 dB	0.4 dB
Silent	0.5 dB	0.4 dB	0.3 dB	0.1 dB	-0.5 dB
Stefan	2.7 dB	2.6 dB	2.6 dB	2.6 dB	2.4 dB
Mobile Calendar	0.1 dB	0.0 dB	0.0 dB	-0.2 dB	-0.8 dB

As said before, some existing techniques (e.g., [14], [15]) reduce the number of feedback requests without imposing constraints on the maximum number of requests. To allow comparison with such techniques, we implemented the solution of Areia et al. [15]. In specific, the architecture described in this paper is used but the techniques in Sect. IV-A, IV-B, IV-C and IV-F are replaced by the techniques described in [15]. Bitplane intra coding is switched off, since [15] does not define WZ rate estimation in such cases. In addition, the techniques in [15] are extended to GOP's larger than two by considering hierarchical WZ layers as for the proposed system (Sect. IV-A2).

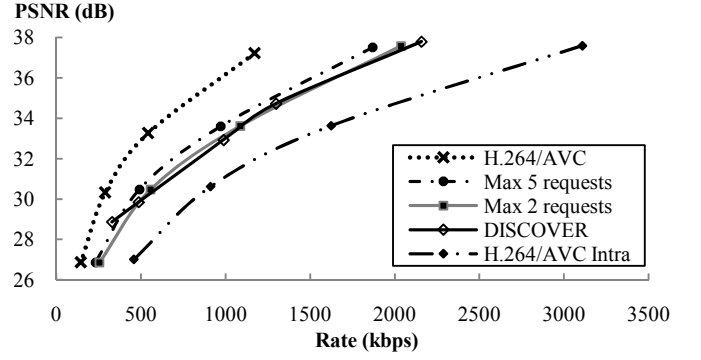
Using the configuration based on [15] (denoted C_{ref} from here on) we recorded the average and maximum number of requests per WZ frame. These results should be compared with a fair configuration of the algorithm proposed in this paper. The main problem here is that the goals of both techniques are not exactly the same. In specific, [15] tries to reduce the average number of requests without caring explicitly about the maximum, whereas our system focuses on the maximum number of requests without explicitly caring about the average. Therefore, defining N based on the maximum or average observed for C_{ref} could not be entirely fair. As a solution, tests with different values of N were conducted, selecting the configuration having the same RD performance as C_{ref} . This way both configurations can be compared by means of the average and maximum number of feedback requests, emphasizing the goals of both approaches.

The results in Table IV indicate that for the same compression performance, compared to C_{ref} the proposed technique offers a significant reduction of the maximum number of feedback requests. Although in most cases the average number of requests is also reduced, this is not always true specifically for sequences with low and/or sudden motion (such as Silent).

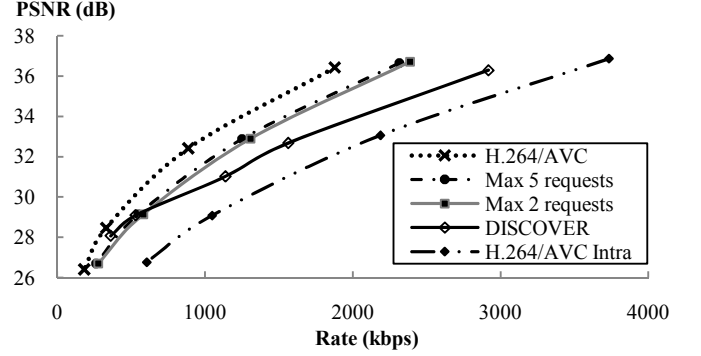
VI. ANALYZING PRACTICAL IMPLICATIONS FOR THE ENCODER AND DECODER

The main motivation for introducing a DVC system supporting constrained feedback was to be more practical than unconstrained solutions currently described in the literature. Therefore, in this section we will analyze practicality starting from an example scenario. The results will be generalized to arbitrary GOP sizes and an arbitrary number of feedback requests N .

Consider the example depicted in Fig. 11, where an encoder receives one frame each t_F seconds, for example, directly



(a) Table Tennis



(b) Coastguard

Fig. 10. RD results (CIF, 30 Hz, GOP 4) of two configurations compared to H.264/AVC intra only, H.264/AVC inter coding, and DISCOVER.

TABLE IV
AVERAGE AND MAXIMUM NUMBER OF REQUESTS FOR THE PROPOSED SYSTEM AND A CONFIGURATION BASED ON [15] (BOTH WZ ONLY). N HAS BEEN DEFINED SO THAT THE AVERAGE BJØNTEGAARD DELTA [25] RATE IS LESS THAN 1%.

	Foreman				Bus			
	Using [15]		Prop.		Using [15]		Prop.	
	avg	max	avg	max	avg	max	avg	max
Q1	7.3	24	5.4	7	6.0	12	3.8	5
Q2	6.1	18	5.0	7	7.1	25	3.8	5
Q3	4.8	22	4.4	7	6.2	38	3.6	5
Q4	3.9	20	3.8	7	3.8	20	3.4	5

	Mobile Calendar				Silent			
	Using [15]		Prop.		Using [15]		Prop.	
	avg	max	avg	max	avg	max	avg	max
Q1	4.7	12	3.7	5	4.7	17	5.7	9
Q2	4.0	9	3.7	5	4.4	17	5.3	9
Q3	3.6	7	3.4	5	3.9	15	4.9	9
Q4	3.7	12	3.2	5	3.0	13	4.3	9

coming from the output of a digital camera. First, each frame is classified as a key frame or WZ frame, according to a GOP of length $K = 4$. The key frames are intra coded and immediately sent to the decoder. For simplicity, we will assume that intra (de)coding requires (at most) t_F seconds. Also, we will assume that the communication delay, denoted t_N (in seconds), is constant.

The WZ frames are transformed and quantized, and the bitplanes are extracted. These bitplanes are stored temporarily in a buffer, in order to wait for the decoder's instructions on

how to code these bitplanes (i.e., the mode to use and the number of WZ bits to send). Meanwhile, the encoder proceeds by processing or buffering the following frame received as input.

At the decoder, intra coded frames are decoded as soon as possible. When I'_5 is available, the decoder starts decoding W_3 . First, side information Y_3 is generated, using I'_1 and I'_5 as references, and the correlation noise is estimated. Denote t_{SI} the maximum time (in seconds) to perform these computations. Using the side information, the correlation, and information about the decoding process for previously decoded WZ frames, the decoder defines the modes as well as the number of WZ bits for each of the N requests, using the techniques described earlier in this paper. The computational effort required in this context is low, and will therefore be neglected.

To evaluate the worst case scenario we will assume that all N requests need to be issued to complete WZ decoding. The first request REQ_1 arrives at the encoder after a transmission delay of t_N seconds. For simplicity, we assume that the encoder is able to send the response RSP_1 without computational overhead. In other words, the decoder receives the response $2t_N$ seconds after REQ_1 has been sent out. Using the information in RSP_1 , the intra bitplanes are decoded, and the turbo decoding (TD) process is started for the WZ bitplanes (if any). Denote t_D the time required for turbo decoding (successful or unsuccessful). In the worst case scenario, all N requests are issued, so that the maximum time required for decoding one WZ frame is approximated by:

$$t_{SI} + N \cdot (2t_N + t_D). \quad (12)$$

When W'_3 is available, the decoder generates the side information for W_2 (using I'_1 and W'_3 as references) and decodes this frame in a similar way by issuing at most N requests for bits. For W_4 remark that – although Y_4 can be generated earlier using W'_3 and I'_5 – the decoder has to wait until bitplane decoding for W'_2 has terminated, before it can issue the first request REQ_1 . This is because, when following the techniques described in this paper, rate estimation and mode decision for W_4 depend on the bitplane rates and decoding success of W'_2 . Remark that there are similar dependencies between other frames, e.g., the first request for W'_6 can only be constructed when W'_4 has been decoded etc.

A. Requirements for real-time output

Once W'_2 is available it can be sent to the output, e.g., for display. A real-time decoder is expected to deliver one decoded frame each t_F seconds. Hence, sending W'_2 to the output defines when other WZ frames will need to be available for output. This poses constraints on the WZ decoding time, and consequently it defines the feasible range for N . For example, the constraint for delivering W'_4 (and in fact all other frames from the lowest WZ layer) is given by $N \cdot (2t_N + t_D) \leq 2t_F$. As a result, N is constrained through the condition:

$$N \leq \frac{2t_F}{2t_N + t_D}. \quad (13)$$

Remark that – due to hierarchical coding – the frame rate of the lowest WZ layer is always the same (i.e. $1/2t_F$) regardless

TABLE V
APPROXIMATE MAXIMUM ROUND TRIP TIME (RTT) FOR SUPPORTING A SPECIFIC NUMBER OF REQUESTS N (FOR 30 HZ SEQUENCES).

N	1	2	3	4	5	6	7	8
max RTT	66.7	33.3	22.2	16.7	13.3	11.1	9.5	8.3

of the GOP length. This means that Eq. 13 is independent from the GOP size.

Using this equation we can calculate the number of requests supported given a particular network delay t_N . For simplicity, t_D has been neglected compared to $2t_N$. In addition, t_F has been taken equal to 33.3 ms (corresponding to 30 Hz sequences). The results are listed in Table V.

Table V shows that the proposed configuration is useful up to a RTT of about 67 ms, since no requests can be issued for RTTs exceeding this threshold. Larger RTTs can be supported, for example, by reducing the frame rate to 15 Hz. Alternatively, extensions to the proposed algorithm could be defined, which is left as a topic of future work.

Table V provides good indications that the proposed DVC system is useful in practice. For example, average round trip latencies in 2010 on AT&T's global IP network was reported to be about 14 ms between city pairs in Europe, 34 ms between pairs in the US, 63 ms for Asia Pacific and 111 ms for Latin America [26]. As we can see, even for non-specialized infrastructures the threshold of 67 ms is often respected, even providing a margin to support multiple requests in many cases. This creates confidence that the proposed techniques can be applied in the typically more specialized DVC application scenarios [27], [28] where latencies are expected to be lower.

For comparison, our configuration implementing the techniques by Areia et al. [15] required up to 38 requests for the Bus sequence (Table IV). This means that – to support the worst case – the network RTT should be not larger than 1.8 ms. Clearly, this limits the applicability of such techniques severely, illustrating the need for feedback-constrained solutions as proposed in this paper.

B. Analyzing end-to-end delay

A second important parameter in video communications is the end-to-end delay, i.e., the time between the reception of a frame at the encoder and its output at the decoder. In our example scenario, the end-to-end delay Δ is given by:

$$5t_F + 9t_N + 2t_{SI} + 4t_D. \quad (14)$$

Generalizing this equation to arbitrary GOPs of length K and arbitrary N results in:

$$\Delta = (K + 1) \cdot t_F + (2N \log_2(K) + 1) \cdot t_N + \log_2(K) \cdot t_{SI} + N \log_2(K) t_D. \quad (15)$$

To facilitate reasoning about the order of magnitude of Δ , the above formula is approximated by neglecting the terms involving t_{SI} and t_D , i.e.:

$$\Delta \approx (K + 1) \cdot t_F + (2N \log_2(K) + 1) \cdot t_N. \quad (16)$$

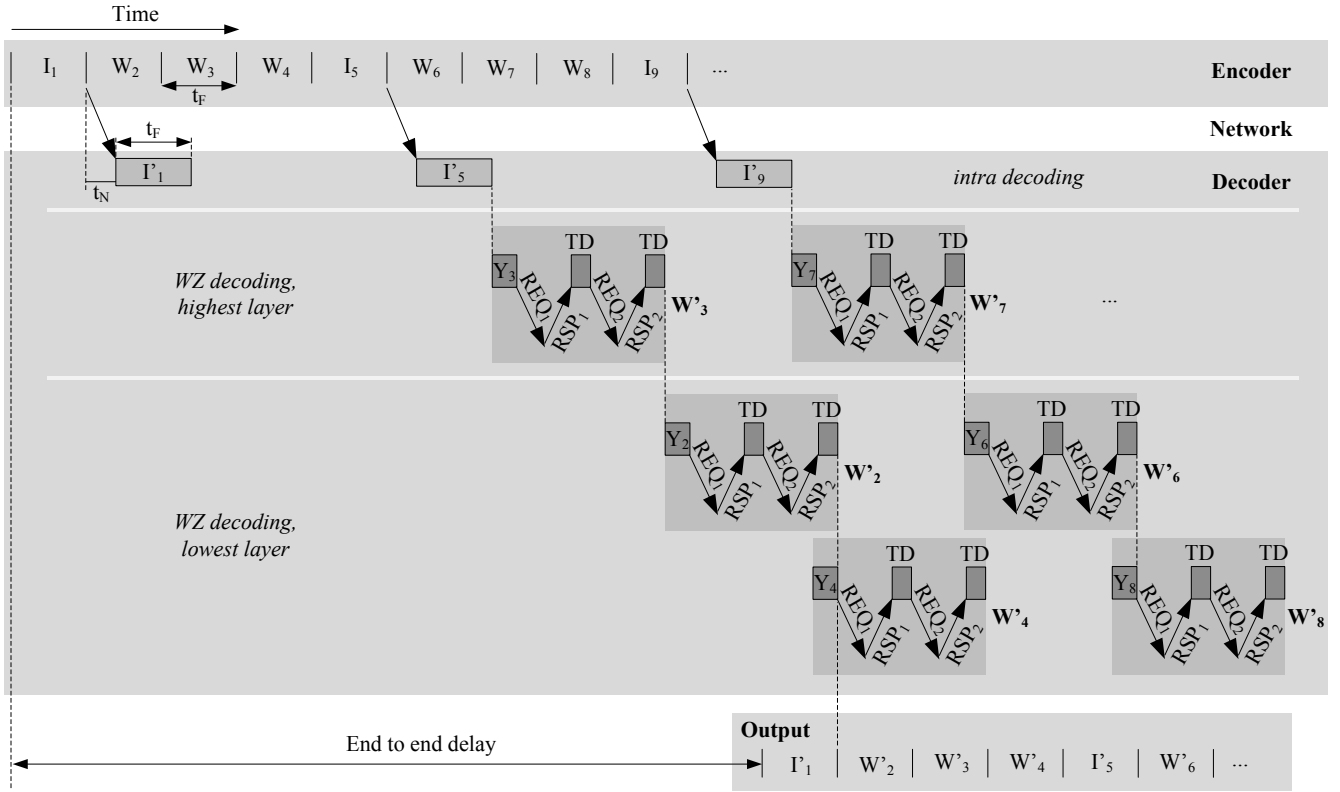


Fig. 11. Illustration of a possible scenario for a DVC system constrained to a maximum of 2 requests.

Using this approximation, Δ is plotted as a function of the round trip time $2t_N$ for different values of N . The GOP size used was four with a frame rate of 30 Hz.

The results depicted in Fig. 12 show end-to-end delays from 167 ms up to 330 ms. Remark that the upper limit on the end-to-end delay is defined by the maximum RTT values defined in Table V. These delays are acceptable for many unidirectional video streaming scenarios, illustrating the practical usefulness of feedback-constrained DVC systems.

Note that the end-to-end delay can be drastically decreased by generating the side information through extrapolation [29] instead of interpolation, as in that case $(K+1) \cdot t_F$ in Eq. 15 would be replaced by $2t_F$.

C. Encoder buffer occupancy

Another issue worth discussing is that – although the encoder does not need to perform additional computations – it needs to store the WZ frames temporarily in a buffer to wait for the decoder's instructions. Given the low requirements of the encoder in terms of complexity, cost and/or size, it is therefore important to model the required size of the WZ frame buffer.

From the example in Fig. 11 we can see that W_3 has to stay in memory at the encoder from the time it is received as input up to the transmission of the final request RSP_2 . This corresponds to a duration of

$$4t_F + t_{SI} + Nt_N + (N-1) \cdot t_D \text{ seconds.} \quad (17)$$

Assuming that the decoder operates in real-time, W_7 needs to be stored for the same duration as well, as well as other

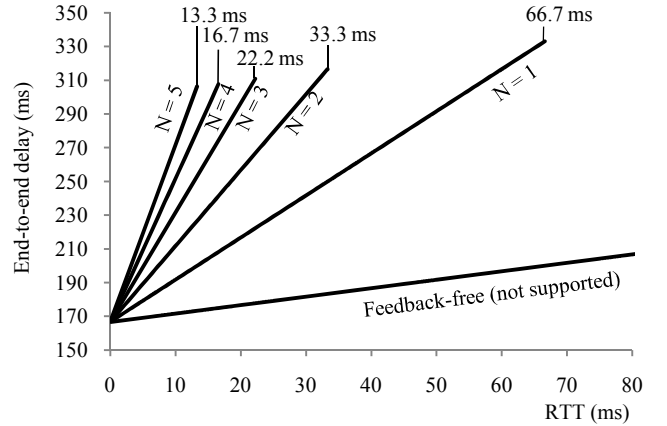


Fig. 12. Approximated end-to-end delay as a function of network round trip time (RTT), for sequences at 30 Hz and a GOP of size four. Curves for supported configurations are provided (i.e., $N = 1$ to 5), as well as for feedback-free DVC systems.

frames at the corresponding position in the following GOPs. Since the rate of these frames is one frame each $4t_F$ seconds, the number of frames to store for the highest WZ layer is given by:

$$\left\lceil \frac{4t_F + t_{SI} + Nt_N + (N-1) \cdot t_D}{4t_F} \right\rceil. \quad (18)$$

Remark that upward rounding is performed here to obtain the required size of the frame buffer instead of its average occupancy.

Similarly, for the first WZ position in each GOP the encoder

has a maximum of

$$\left\lceil \frac{5t_F + 2t_{SI} + 2Nt_N + (2N - 1) \cdot t_D}{4t_F} \right\rceil \quad (19)$$

frames to store, whereas for the final WZ position

$$\left\lceil \frac{3t_F + 2t_{SI} + 3Nt_N + (3N - 1) \cdot t_D}{4t_F} \right\rceil \quad (20)$$

frames need to be stored.

Given these equations, the total size of the encoder's WZ buffer is the sum of the three terms given by Eq. 18, Eq. 19, and Eq. 20. While this analysis applies to a GOP of size four only (but arbitrary N), similar reasoning applies to different GOP lengths.

Similarly to previous examples, the terms involving t_{SI} and t_D have been neglected, delivering a required total WZ buffer capacity of five or six frames (depending on the RTT). The upper bound of six frames is imposed by the real-time constraint defined previously by Eq. 13.

VII. CONCLUSIONS AND FUTURE WORK

This paper provided a first study on how to constrain the number of feedback requests in a DVC system to a fixed value. By constraining the number of requests several important parameters are constrained as well, including the system's end-to-end delay and the encoder's buffer size. Analyzing these properties provided clear indications that the proposed approach is useful in practical streaming scenarios.

The penalty in compression performance has shown to be negligible when at least five feedback requests can be supported. In addition, due to improvements such as mode decision, the system is able to achieve similar or better performance than DISCOVER even when constrained to only two requests per WZ frame.

When a limited form of feedback can be supported, the proposed solution has several advantages over feedback-free DVC systems. The main advantage is that there is no impact on the computational complexity of the encoder, since the decoder is responsible for rate estimation and mode decision. Also, compression performance can be improved by only modifying the decoder, which could be a considerable advantage in the context of deployment or standardization.

Apart from improving the proposed techniques, an important topic for further research includes refining the system analysis. In this paper we have occasionally neglected the time required for generating side information and turbo decoding. However, these calculations should be taken into account to allow a more accurate system evaluation.

ACKNOWLEDGMENT

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

REFERENCES

- [1] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conference (DCC)*, pp. 252–261, April 2002.
- [2] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proc. SPIE Visual Communications and Image Processing*, vol. 5308, pp. 520–528, January 2004.
- [3] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," in *Proc. IEEE, Special Issue on Video Coding and Delivery*, vol. 93, pp. 71–83, January 2005. Invited Paper.
- [4] S. Sofke, F. Pereira, and E. Müller, "Dynamic quality control for transform domain wyner-ziv video coding," *EURASIP Journal on Image and Video Processing*, vol. 2009, pp. 1–15, 2009.
- [5] R. Martins, C. Brites, J. Ascenso, and F. Pereira, "Refining side information for improved transform domain wyner-ziv video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 1327–1341, september 2009.
- [6] X. Fan, O. Au, and N. M. Cheung, "Transform-domain adaptive correlation estimation (TRACE) for Wyner-Ziv video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 1423–1436, November 2010.
- [7] J. Slowack, J. Škorupa, S. Mys, P. Lambert, C. Grecos, and R. Van de Walle, "Flexible distribution of complexity by hybrid predictive-distributed video coding," *Signal Processing: Image Communication*, vol. 25, pp. 94–110, February 2010.
- [8] F. Dufaux and T. Ebrahimi, "Encoder and decoder side global and local motion estimation for distributed video coding," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, October 2010.
- [9] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," in *Proc. Picture Coding Symposium (PCS)*, November 2007.
- [10] M. Morbée, J. Prades-Nebot, A. Pizurica, and W. Philips, "Feedback channel suppression in pixel-domain Distributed Video Coding," in *Annual Workshop on Circuits, Systems and Signal Processing (ProRISC)*, pp. 154–157, November 2006.
- [11] C. Brites and F. Pereira, "Encoder rate control for transform domain Wyner-Ziv coding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. II–5–II–8, September 2007.
- [12] C. Fu and J. Kim, "Encoder rate control for block-based distributed video coding," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 333–338, October 2010.
- [13] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conference on Communication, Control and Computing*, October 2002.
- [14] D. Kubasov, K. Lajnef, and C. Guillemot, "A hybrid encoder/decoder rate control for a Wyner-Ziv video codec with a feedback channel," in *IEEE MultiMedia Signal Processing Workshop*, pp. 251–254, October 2007.
- [15] J. D. Areia, J. Ascenso, C. Brites, and F. Pereira, "Low complexity hybrid rate control for lower complexity wyner-ziv video decoding," in *16th European Signal Processing Conference*, August 2008.
- [16] J. Slowack, S. Mys, J. Škorupa, N. Deligiannis, P. Lambert, A. Munteanu, and R. Van de Walle, "Rate-distortion driven decoder-side bitplane mode decision for distributed video coding," *Signal Processing: Image Communication*, vol. 25, pp. 660–673, October 2010.
- [17] J. Slowack, S. Mys, J. Škorupa, P. Lambert, C. Grecos, and R. Van de Walle, "Accounting for quantization noise in online correlation noise estimation for distributed video coding," in *Proc. Picture Coding Symposium (PCS)*, May 2009.
- [18] Y. Vatis, S. Klomp, and J. Ostermann, "Inverse bit plane decoding order for turbo code based Distributed Video Coding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. II–1–II–4, September 2007.
- [19] W. Weerakkody, W. Fernando, J. Martinez, P. Cuenca, and F. Quiles, "An iterative refinement technique for side information generation in dvc," in *IEEE International Conference on Multimedia and Expo*, pp. 164–167, July 2007.
- [20] S. Ye, M. Ouaret, F. Dufaux, and T. Ebrahimi, "Improved side information generation with iterative decoding and frame interpolation for distributed video coding," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2228–2231, October 2008.
- [21] X. Huang and S. Forchhammer, "Transform domain wyner-ziv video coding with refinement of noise residue and side information," in *Proc. of SPIE*, vol. 7744, pp. 774418–1–774418–9, July 2010.

- [22] X. Fan, O. C. Au, N. M. Cheung, Y. Chen, and J. Zhou, "Successive refinement based Wyner-Ziv video compression," *Signal Processing: Image Communication*, vol. 25, pp. 47–63, January 2010.
- [23] C. Brites, J. Ascenso, J. Q. Pedro, and F. Pereira, "Evaluating a feedback channel based transform domain wyner-ziv video codec," *Signal Processing: Image Communication*, pp. 269–297, 2008.
- [24] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in Wyner-Ziv video coding with multiple side information," in *IEEE MultiMedia Signal Processing Workshop*, pp. 183–186, October 2007.
- [25] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," April 2001. VCEG Contribution VCEG-M33.
- [26] AT&T, "Global IP network latency averages." http://ipnetwork.bgtmo.ip.att.net/pws/global_network_avgs.html. Accessed: January 10, 2011.
- [27] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp, "Distibuted video coding: Selecting the most promising application scenarios," *Signal Processing: Image Communication*, pp. 339–352, 2008.
- [28] N. Deligiannis, F. Verbist, J. Barbarien, J. Slowack, R. Van de Walle, P. Schelkens, and A. Munteanu, "Distributed coding of endoscopic video," in *Proceedings IEEE International Conference on Image Processing (ICIP)*, pp. 1853–1856, September 2011.
- [29] S. Borchert, R. Westerlaken, R. K. Gunnewiek, and R. Lagendijk, "On extrapolating side information in Distributed Video Coding," in *Proc. of Picture Coding Symposium (PCS)*, November 2007.